БЛОЧНЫЕ ШИФРЫ

Сеть Фейстеля

Сеть Фейстеля – это один из методов построения блочных шифров, который включает в себя выполнение определенных операций над данными и ключом в каждой ячейке Фейстеля. Все ячейки работают одинаково, и ключ меняется при переходе от одной ячейки к другой. В результате шифрования и расшифрования выполняются одни и те же операции, но с разным порядком ключей. Сеть Фейстеля используется во многих блочных шифрах, таких как DES, RC2, RC5 и других. Схема алгоритма представлена на рис. 28:

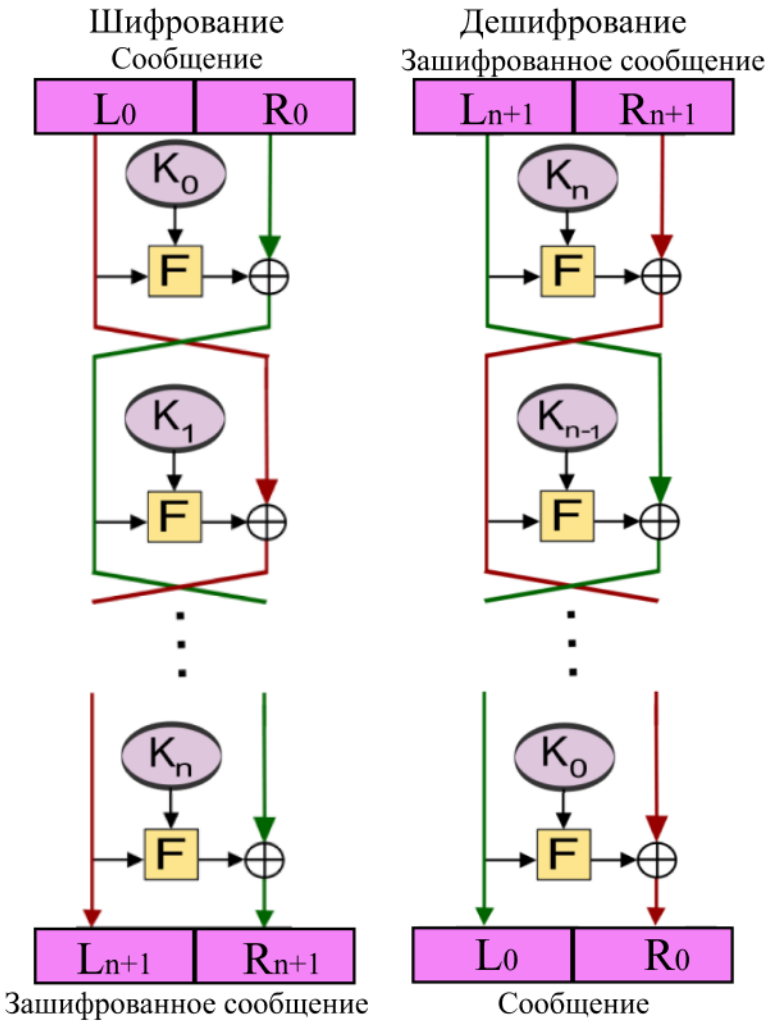

Рисунок 28. Схема алгоритма сети Фейстеля

Достоинства алгоритма:

– простота реализации: сеть Фейстеля относительно проста для понимания и реализации, как в программном, так и в аппаратном обеспечении;

– эффективность: благодаря своей простоте, сеть Фейстеля может быть эффективной с точки зрения скорости работы и использования ресурсов;

– гибкость: сеть Фейстеля позволяет использовать различные методы шифрования, что делает ее гибкой и адаптируемой к различным задачам.

Недостатки алгоритма:

– сеть Фейстеля не является особенно устойчивой к криптоаналитическим атакам;

– сложность анализа: анализ безопасности сети Фейстеля может быть сложным из-за ее итерированной структуры и использования ключей.

Пример

Зашифровать сообщение используя алгоритм сети Фейстеля.

Сообщение: КТСО

Ключ: ЛЕТО

Известно, что функцией F данного алгоритма является операция XOR с ключом.

Таблица 33. Алфавит для Сети Фейстеля

№	Буква	Двоичный код	№	Буква	Двоичный код	№	Буква	Двоичный код
1	А	00000001	12	К	00001100	23	Х	00010111
2	Б	00000010	13	Л	00001101	24	Ц	00011000
3	В	00000011	14	М	00001110	25	Ч	00011001
4	Г	00000100	15	Н	00001111	26	Ш	00011010
5	Д	00000101	16	О	00010000	27	Щ	00011011
6	Е	00000110	17	П	00010001	28	Ъ	00011100
7	Ё	00000111	18	Р	00010010	29	Ы	00011101
8	Ж	00001000	19	С	00010011	30	Ь	00011110
9	З	00001001	20	Т	00010100	31	Э	00011111
10	И	00001010	21	У	00010101	32	Ю	00100000
11	Й	00001011	22	Ф	00010110	33	Я	00100001

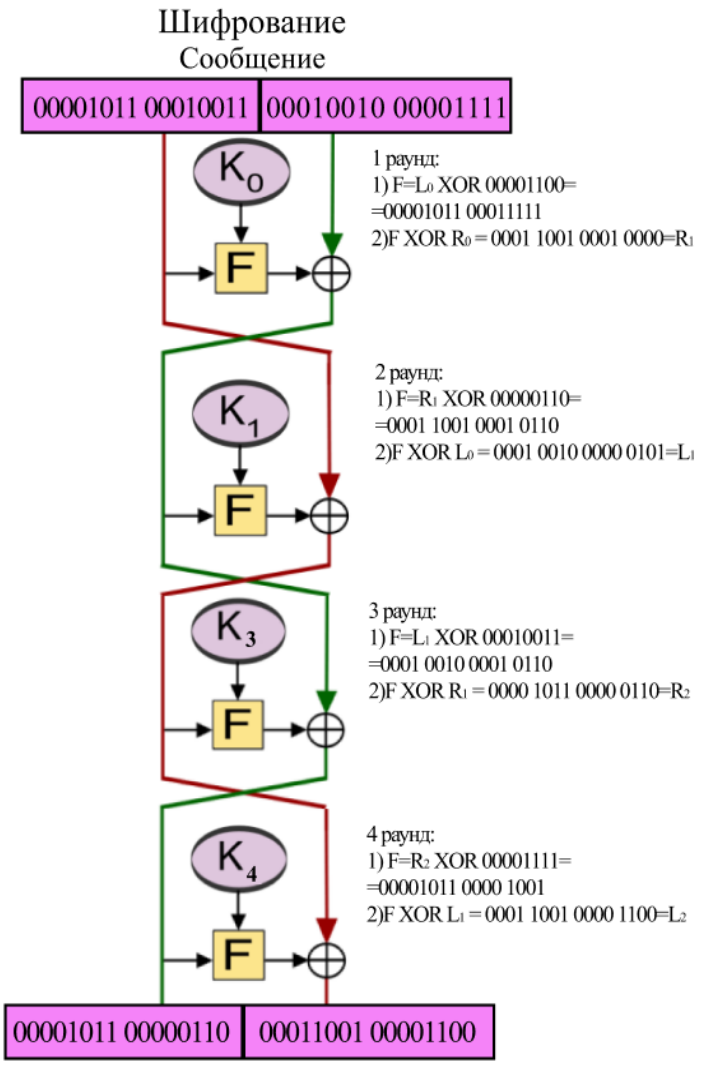

Рисунок 29. Пример задачи

Ответ: зашифрованное сообщение - КЕШЛ

Задача: Имея зашифрованное сообщение и ключ, получить исходное сообщение. Зашифрованное сообщение приведено в 10 с.с.. Каждое число весит 8 бит.

Вариант 1

Зашифрованное сообщение	13 00 45 26
Ключ	КТСО

Вариант 2

Зашифрованное сообщение	12 04 02 44
Ключ	ЗАРЯ

Вариант 3

Зашифрованное сообщение	13 01 28 19
Ключ	КИИБ

Вариант 4

Зашифрованное сообщение	17 20 29 13
Ключ	ЛИСА

Вариант 5

Зашифрованное сообщение	08 17 05 08
Ключ	ЗИМА

Вариант 6

Зашифрованное сообщение	25 28 06 25
Ключ	АВТО

Вариант 7

Зашифрованное сообщение	01 00 18 10
Ключ	БЛОК

Вариант 8

Зашифрованное сообщение	01 06 20 07
Ключ	ЗВУК

Вариант 9

Зашифрованное сообщение	15 29 01 30
Ключ	ОКНО

Вариант 10

Зашифрованное сообщение	12 22 31 48
Ключ	ЛЕТО

ГОСТ 28147-89 (Магма)

ГОСТ 28147-89 был опубликован под названием «Магма» как часть стандарта ГОСТ Р 34.12-2015, а позже как часть стандарта ГОСТ 34.12-2018. В 2020 году алгоритм «Магма» был опубликован в виде RFC 8891

ГОСТ 28147-89 — блочный шифр с 256-битным ключом и 32 циклами (называемыми раундами) преобразования, оперирующий 64-битными блоками. Основа алгоритма шифра — сеть Фейстеля.

Выделяют четыре режима работы ГОСТ 28147-89:

1) простой замены;
2) гаммирование;
3) гаммирование с обратной связью;
4) режим выработки имитовставки.

RC5

RC5 (Ron’s Code 5 или Rivest’s Cipher 5) — это блочный шифр, разработанный Роном Ривестом из компании RSA Security с переменным количеством раундов, длиной блока и длиной ключа. Это расширяет сферу использования и упрощает переход на более сильный вариант алгоритма.

Существует несколько различных вариантов алгоритма, в которых преобразования в «пол-раундах» классического RC5 несколько изменены. В классическом алгоритме используются три примитивных операции и их инверсии:

1) сложение по модулю;
2) побитовое исключающее ИЛИ (XOR);
3) операции циклического сдвига на переменное число бит.

Основным нововведением является использование операции сдвига на переменное число бит, не использовавшиеся в более ранних алгоритмах шифрования. Эти операции одинаково быстро выполняются на большинстве процессоров, но в то же время значительно усложняют дифференциальный и линейный криптоанализ алгоритма.

Шифрование по алгоритму RC5 состоит из двух этапов. Процедура расширения ключа и непосредственно шифрование. Для расшифрования выполняется сначала процедура расширения ключа, а затем операции, обратные процедуре шифрования. Все операции сложения и вычитания выполняются по модулю.

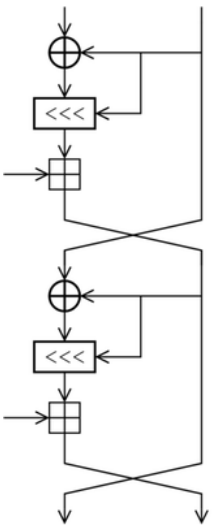

Рисунок 30. Схема алгоритма шифрования RC5

Поскольку RC5 имеет переменные параметры, для обозначения алгоритма с определенными параметрами используется обозначение RC5-W/R/b, где:

W - половина размера блока в битах. Возможные значения 16, 32 и 64. Рекомендуется выбирать W равным размеру машинного слова для эффективной реализации. Например, для 32-битных систем оптимальным будет W = 32 (размер блока 64 бита);

R - количество раундов. Может принимать значения от 0 до 255. Увеличение R повышает уровень безопасности шифра. При R = 0 шифрование не производится. В алгоритме RC5 также используется таблица расширенных ключей размером 2(R + 1) слова, которая генерируется из заданного пользователем ключа;

b - длина ключа в байтах. Может принимать значения от 0 до 255.

Пример

Для шифрования используется упрощенная схема RC4, представленная на рис. 31.

Операция «сложение» в данном примере выполняется не по модулю.

Пример

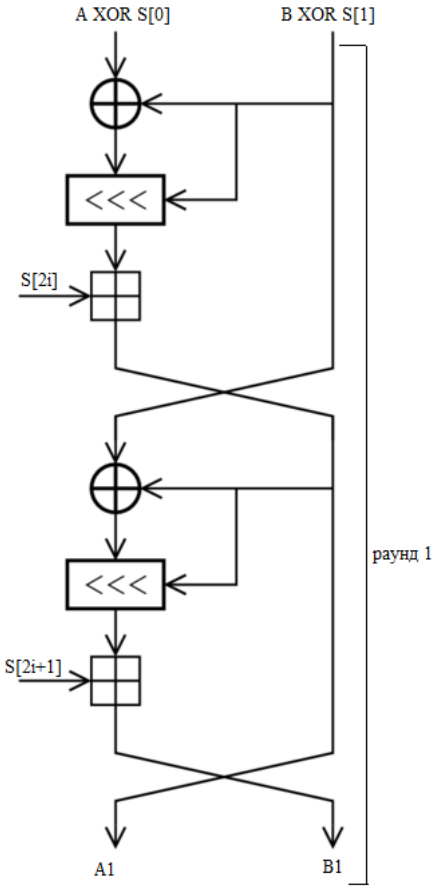

Рисунок 31. Упрощенная схема RC4|У

Сообщение: ЛЕТО

Л	00000001	A
Е	00000110	
Т	00010011	B
О	00001111	

Ключ: КНИГА

К	00001011	S[0]
Н	00001110	S[1]
И	00001001	S[2]
Г	00000100	S[3]
А	00000001	S[4]

Таблица 34. Двоичный код букв

№	Буква	Двоичный код	№	Буква	Двоичный код	№	Буква	Двоичный код
1	А	00000001	12	К	00001100	23	Х	00010111
2	Б	00000010	13	Л	00001101	24	Ц	00011000
3	В	00000011	14	М	00001110	25	Ч	00011001
4	Г	00000100	15	Н	00001111	26	Ш	00011010
5	Д	00000101	16	О	00010000	27	Щ	00011011
6	Е	00000110	17	П	00010001	28	Ъ	00011100
7	Ё	00000111	18	Р	00010010	29	Ы	00011101
8	Ж	00001000	19	С	00010011	30	Ь	00011110
9	З	00001001	20	Т	00010100	31	Э	00011111
10	И	00001010	21	У	00010101	32	Ю	00100000
11	Й	00001011	22	Ф	00010110	33	Я	00100001

Количество раундов r=1, следовательно i=1..r.

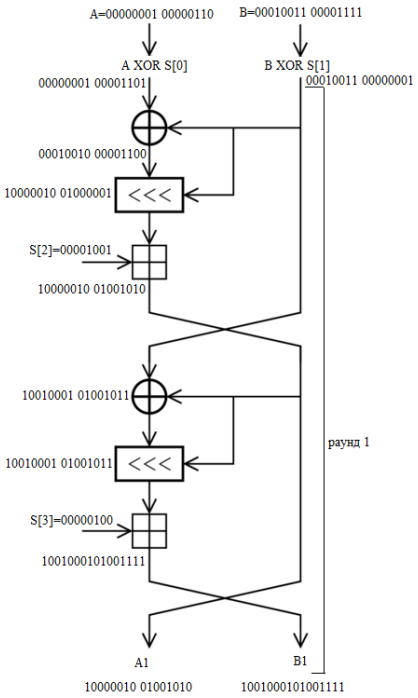

Рисунок 32. Пример шифрования

Пояснение для циклического сдвига с двумя переменными:

10010001 01001011<<<10000010 01001010 – циклический сдвиг влево на 16

10000010 01001010 – 16 значащих битов

10010001 01001011 циклический сдвиг на 16 = 10010001 01001011

Зашифрованное сообщение:

A1	10000010	Б
	01001010	Й
B2	10010001	Р
	01001111	О

Примечание:

Первые 3 бита полученной зашифрованное буквы заменяем 0.

RC6

RC6 – симметричный блочный криптографический алгоритм, производный от алгоритма RC5. Был создан Роном Ривестом, Мэттом Робшау и Рэем Сиднеем для удовлетворения требований конкурса Advanced Encryption Standard (AES). Алгоритм был одним из пяти финалистов конкурса, был также представлен NESSIE и CRYPTREC. Является собственническим (проприетарным) алгоритмом, и запатентован RSA Security.

Для спецификации алгоритма с конкретными параметрами, принято обозначение RC6-w/r/b, где

- w — длина машинного слова в битах;
- r — число раундов;
- b — длина ключа в битах. Возможные значения 0..255 бит.

Схема одного раунда в алгоритме RC6 выглядит так:

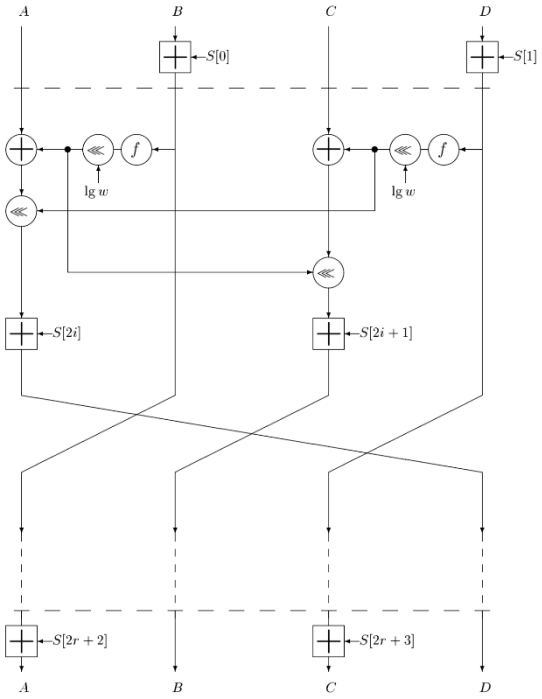

Рисунок 33. Схема алгоритма RC6

Генерация констант:

Так же, как и в RC5, в RC6 генерируются две псевдослучайные величины, используя две математические константы: экспонента (e) и золотое сечение (f).

$$Q_w \leftarrow Odd((f-1)*2^w);$$
$$P_w \leftarrow Odd((e-1)*2^w),$$

где Odd() — это округление до ближайшего нечетного целого. При w = 32 бита (в шестнадцатеричном виде):

P_{32}=9E3779B9

Q_{32}=B7E15163

В десятичном виде:

P_{32}=3084996963

Q_{32}=2654435769

Процедура расширения ключа для RC6-w/r/b:

Таблица ключей для шифра RC6 также идентична таблице шифра RC5. Отличие состоит в том, что большее количество слов из массива L получено из

предоставленного пользователем ключа для использования в течение шифрования и расшифровки.

Вход:

b-байтный ключ, заданный пользователем, предварительно преобразованный в массив из слов L[0,…,c-1].

r — количество раунд

w-битная таблица ключей S[0,…,2r+3] .

Выход:

>>>, <<< - циклические сдвиги вправо и влево соответственно.

Пример.

Для шифрования используется упрощенная схема RC4, представленная на рис. 34.

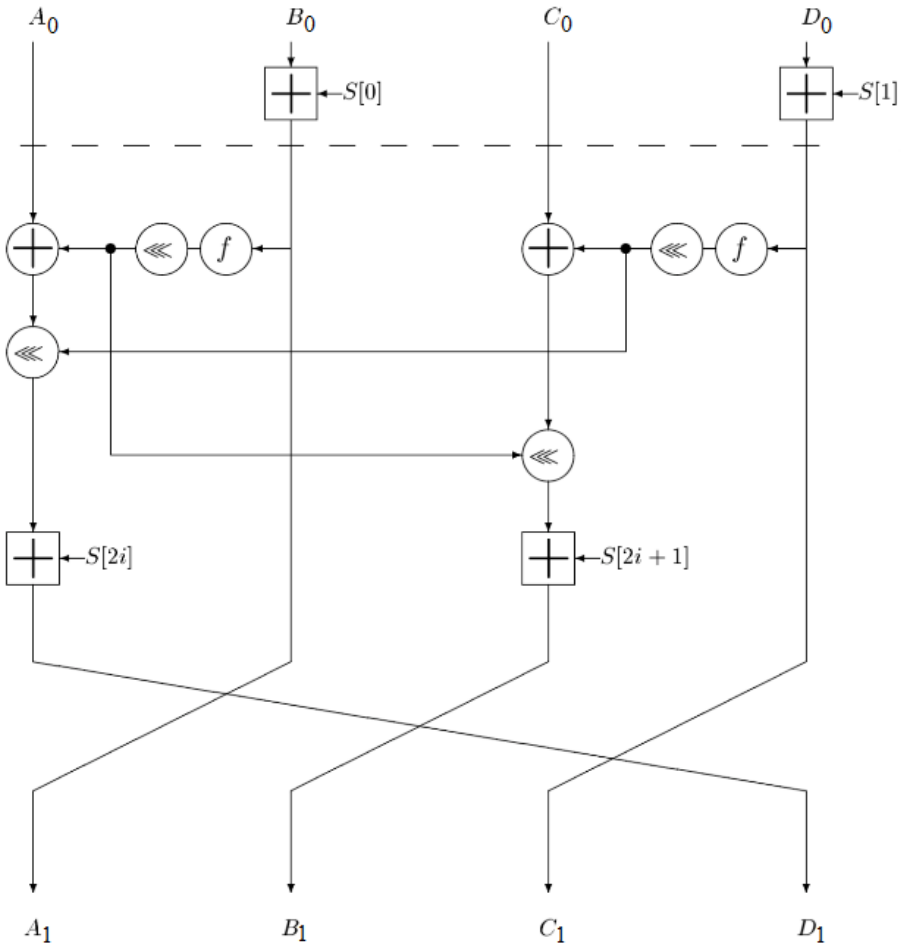

Рисунок 34. Упрощенная схема алгоритма RC6\У

Сообщение: ЛЕТО

Л	00001101	A0
Е	00000110	B0
Т	00010100	C0
О	00010000	D0

Ключ: КНИГА

К	00001100	S[0]
Н	00001111	S[1]
И	00001010	S[2]
Г	00000100	S[3]
А	00000001	S[f]

Таблица 35. Двоичный код букв

№	Буква	Двоичный код	№	Буква	Двоичный код	№	Буква	Двоичный код
1	А	00000001	12	К	00001100	23	Х	00010111
2	Б	00000010	13	Л	00001101	24	Ц	00011000
3	В	00000011	14	М	00001110	25	Ч	00011001
4	Г	00000100	15	Н	00001111	26	Ш	00011010
5	Д	00000101	16	О	00010000	27	Щ	00011011
6	Е	00000110	17	П	00010001	28	Ъ	00011100
7	Ё	00000111	18	Р	00010010	29	Ы	00011101
8	Ж	00001000	19	С	00010011	30	Ь	00011110
9	З	00001001	20	Т	00010100	31	Э	00011111
10	И	00001010	21	У	00010101	32	Ю	00100000
11	Й	00001011	22	Ф	00010110	33	Я	00100001

Количество раундов r=1, следовательно i=1..r.

Пусть функция f отвечает за результат операции «побитовое сложение» с

ключом S[f].

Циклический сдвиг 1.

Раунд 1: (r=1, i=1)

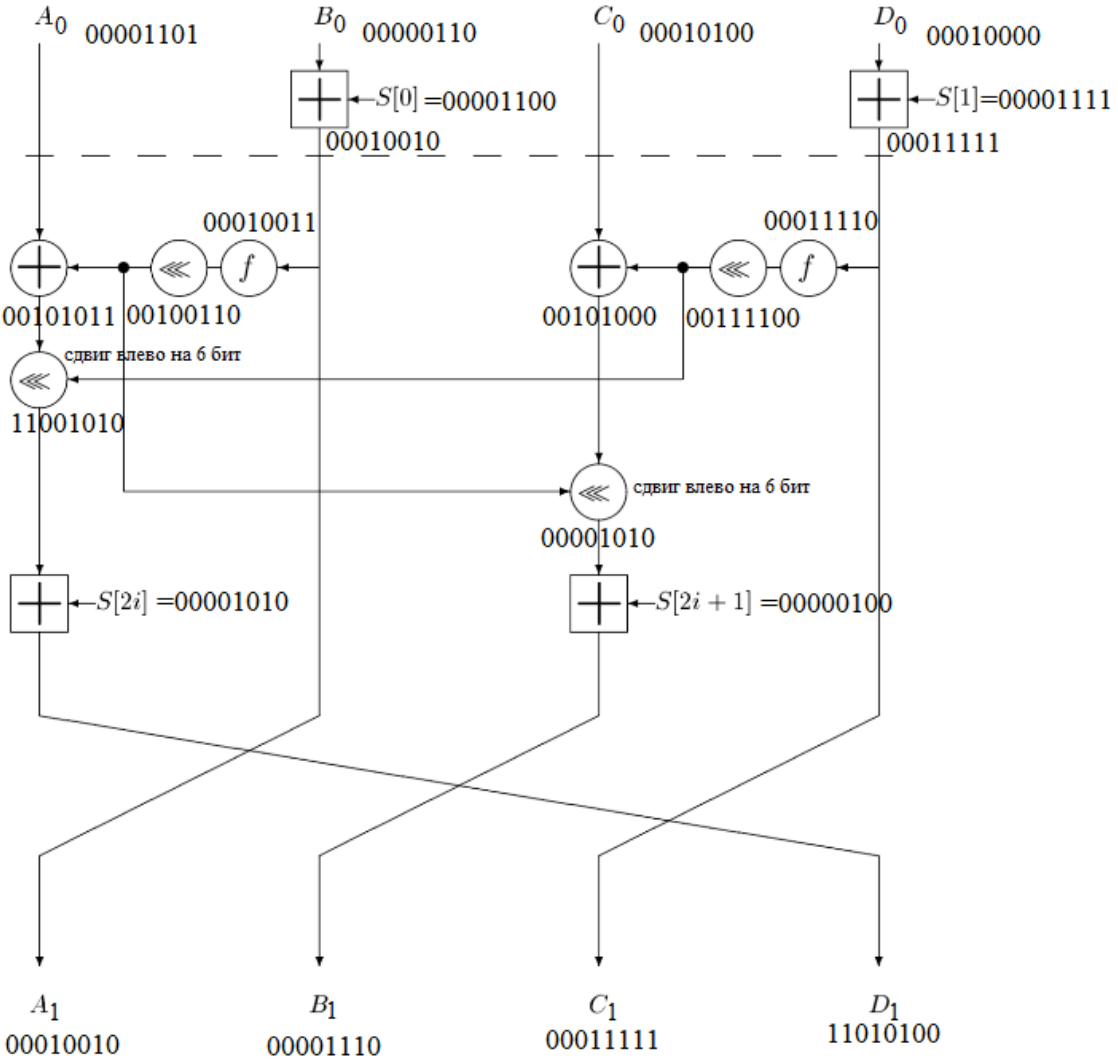

Рисунок 35. Пример шифрования

Пояснение для циклического сдвига с двумя переменными:

00101100<<<00111000 – циклический сдвиг влево на 6

00111000 – 6 значащих битов

00101100 циклический сдвиг на 6 = 00001011

Зашифрованное сообщение:

A_1	00010010	Р
B_1	00001110	М
C_1	00011111	Э
D_1	11010100	Т

Примечание:

Первые 3 бита полученной зашифрованное буквы заменяем 0.

Задача: Имея зашифрованное сообщение и ключ, получить исходное сообщение, количество раундов r=1.

Вариант 1

Сообщение	СВЕТ
Ключ	ЛАМПА
Циклический сдвиг	1

Вариант 2

Сообщение	ЗИМА
Ключ	ОСЕНЬ
Циклический сдвиг	2

Вариант 3

Сообщение	СТОЛ
Ключ	КАРТА
Циклический сдвиг	1

Вариант 4

Сообщение	КТСО
Ключ	УЧЁБА
Циклический сдвиг	2

Вариант 5

Сообщение	ККСО
Ключ	ЦЕНТР
Циклический сдвиг	1

Вариант 6

Сообщение	ФАЙЛ
Ключ	ВИРУС
Циклический сдвиг	1

Вариант 7

Сообщение	УРОК
Ключ	АДРЕС
Циклический сдвиг	1

Вариант 8

Сообщение	МОРЕ
Ключ	АРХИВ
Циклический сдвиг	1

Вариант 9

Сообщение	ПАРА
Ключ	АГЕНТ
Циклический сдвиг	1

Вариант 10

Сообщение	КУРС
Ключ	ХАКЕР
Циклический сдвиг	1

«Кузнечик»

ГОСТ Р 34.12-2015 или «Кузнечик» – это симметричный алгоритм блочного шифрования с размером блока 128 бит и длиной ключа 256 бит, использующий для генерации раундовых ключей SP-сеть.